An Efficient Approach to Automatic Generation of Time-lapse Video Sequences

An Efficient Approach to Automatic Generation of Time-lapse Video Sequences

Javier Calero de Torres[1]; Bryan Gardiner[2]; Ilias Dahi[1]; Sandra Moffett[2]; Marco Herbst[1]; Joan Condell[2]

[1]Evercam LTD, Dublin, Ireland; [2]School of Computing, Engineering & Intelligent Systems, Ulster University

Abstract

Time-lapse video sequences have recently become a highly utilised asset for marketing and advertising, particularly within the field of construction and landscape development. However, the manual generation of these videos, at a quality that can be used for marketing purposes, can be quite time-consuming. In this paper, a novel application for generating time-lapse videos is proposed, which will automatically select the optimal frames for time-lapse video generation, enhance these frames by applying a number of image pre-processing and machine learning techniques such as FAST super-resolution to improve the frames quality, and finally, provide an intuitive user interface to allow users to customise the time-lapse video with company branding. The auto-generated time-lapse videos will use techniques such as Laplacian filtering and temporal smoothing filtering to determine inactivity within the video sequence, classify day or night and, by use of optical character recognition, have the ability to remove unwanted artefacts such as the captured video date and time stamp. The obtained results from the proposed approach produce comparable video sequences to those produced manually, but with the advantage of being generated much faster and not requiring specialised video editing skills to complete.

Keywords: Time-lapse video generation, timestamp removal, FAST super-resolution processing

1 Introduction

Time-lapse video is a technique where a sequence of still images are captured and combined to show the passage of time over a significantly shorter period. The process of creating time-lapse video sequences has always been a creative process in which highly skilled video editors have taken ownership of this task. For example, consider the popular use case of generating time-lapse videos for construction sites; example frames selected from time-lapse video are provided in Figure 1. It has been perceived that the generation of these video sequences is a monotonous task in which the same techniques are continually repeated to produce a high-quality time-lapse video sequence. In addition, capturing data for time-lapse video generation can require an overwhelming amount of data. For example, a single imaging device that captures an image every three seconds will produce 28,800 images per day, equating to over ten million images per year. Image or video compression reduces the storage requirements, but the resulting data has compression artefacts and is not very useful for further analysis. This is further complicated by the inclusion of external factors such as the introduction of image noise, image quality due to camera movement, lighting conditions, etc.

Figure 1: Selected frames from time-lapse video (Moey, 2019)

In this paper, a pipeline is proposed to overcome these aforementioned issues by developing an efficient mechanism for automatically generating time-lapse videos based on the raw captured video data. The approach will

 ISBN 978-0-9934207-4-0

eliminate the need for skilled video editors to spend excessive amount of time manually generating time-lapse videos. Furthermore, the automated analysis of the captured data will allow the system to intelligently select the most appropriate frames to be used for the sequence and enhance these selections to further improve the quality of the final output. This approach can be divided into three key stages: 1) efficient processing environment conversion; 2) frame analysis; 3) frame enhancement. The time-lapse video can then be rendered based on the optimal selection and enhancement of frames. Throughout the development of the system, data has been collected from various cameras located at numerous construction sites, which is subsequently used to test the robustness of the developed system using different viewpoints, backgrounds and scene compositions.

2 Previous Work

Advances in digital technology have improved the efficacy, cost, and benefits of time-lapse videos, making the method an easier, more efficient, and readily available tool. Image analysis utilising image processing software, such as ESRI, Photoshop, and ImageJ, as examples, has provided the ability to extract detailed data and numerical information from colour channels, pixels, and measurements. Numerous specific algorithmic developments for time-lapse video processing has also been proposed. Matusik et al. [2004] used time-lapse data to compute the reflectance field (or light transport) of a scene for a fixed viewpoint. They represent images as a product of the reflectance field and the incoming illumination. However, the method requires estimating the incident illumination using a light probe camera, and the estimated reflectance field combines the effects of reflectance and shadows, which is inefficient. Koppal and Narasimhan [2006] acquire image sequences with a random moving light source. This permits the authors to semi-cluster the images into regions that have similar normals. These generated clusters are then used as priors to bootstrap a variety of vision algorithms, however not as a fully automated solution. Microsoft presented an algorithm [Joshiet. al., 2015] to create hyper-lapse videos that could be executed in real-time HD videos where there is a high frequency of camera movement. Their approach uses a software camera stabilisation system, selecting the frames of the video that best adjust to the desired speed for the resulting video, making the movement of the camera smooth. Xiong et. al., [2018] generates time-lapse videos using multi-stage dynamic generative adversarial networks (MD-GAN). Although this method utilises high-resolution dynamic videos of sky scenes to train the MD-GAN, it requires an image input of 128x128 resolution to generate future realistic time-lapse frames. The pipeline proposed in this paper will enhance existing work by offering a pipeline that will fully automate the generation of high-quality image sequences for high end time-lapse videos fit for company marketing and advertising requirements.

3 Proposed Approach

3.1 Overview

This work is based on data collected from a number of Hikvision HD cameras, which captures consecutive HD images (1920x1080) at a maximum frame rate of 12 frames per second and stored in a cloud-based server. The pipeline is executed on a virtual machine with a fixed IP, composed by 8GB of RAM, 40GB of disk local and two VCPU. The process of generating time-lapse video sequences is outlined in the section and can be divided into three key stages. Section 3.2 outlines the first of these stages, where FAST Super-resolution processing is used to provide an environment that will permit efficient frame processing. The second stage (Section 3.3) is where each frame is analysed and classified according to three factors: day/night detection, detection of blurred images and activity detection. The third stage (Section 3.4) discusses both timestamp removal and temporal smoothing to ensure smooth transition between frames.

3.2 Stage I: FAST Super-resolution Processing

Time-lapse creation is an intensive process. Retrieving images from the cloud, applying temporal smoothing, then date removal can be quite computationally expensive. To reduce computation when processing each frame, it is

 ISBN 978-0-9934207-4-0

proposed to firstly reduce the resolution of each image, apply the enhancement algorithms on the reduced resolution image, then upscale the image back to its original resolution, ultimately improving system performance. An efficient approach to upscaling each frame is by utilising FAST (Free Adaptive Super-resolution via Transfer), a super-resolution convolutional neural network with a sub-pixel model compensation to improve the resolution of images. FAST accelerates algorithms by up to 15x with a visual quality loss of 0.2dB [Zhang et al., 2017].

The FAST framework transfers super-resolution pixels using motion compensation, exploiting adaptive transfer to retain visual quality and applies a super-resolution structure to non-overlapping blocks and removal of block artefacts with an adaptive deblocking filter. In this paper, each frame is first compressed using JPEG compression. Any processing is computed on the compressed version of the image, then FAST is used to upscale the image back to its original resolution with minimal loss of information. To demonstrate the capability of the FAST algorithm, Figure 2 presents the results of upscaling an image of resolution 480x320 to three times its resolution 1140x960, depicted by Image (a) and Image (b) respectively. This demonstrates how compressing an image for efficient processing can be accurately upscaled to its original resolution using this approach.

Figure 2: Illustration of image upscaling using FAST algorithm

3.3 Stage II: Frame Analysis

When generating time-lapse video sequences, it is important to determine what frames in the original video sequence are of sufficient quality to use and which are deemed insufficient. To accomplish this, three key techniques have been identified to classify frames as *usable* or *unusable*. These techniques are blur detection, frame inactivity and day-night detection. If an image is deemed as *usable*, it will be forwarded to Stage III, Frame Enhancement, otherwise it will be deemed as *unusable* and disregarded from the original video sequence.

3.3.1 Blur detection

When capturing dynamic outdoor scenes, it is not uncommon for a number of frames to become blurred. For example, during rainy days, the lens of the camera can be covered by water drops, and therefore introduce blurriness to the captured frame. These frames should be detected and removed from the final time-lapse video sequence. To solve this, a 3x3 Laplacian filter is convolved with each frame to determine what level of blur is present in the captured image. A Laplacian filter is a 2D isotropic measure of the 2nd derivative of an image, which will determine

 ISBN 978-0-9934207-4-0

the variance of high frequency components present in the scene. The variance can be used as a score to determine the blurriness of the image. The Laplacian operator is defined by:

$$Laplace(f) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial \mathrm{y}^2} \quad (1)$$

If the measured variance within a scene falls below a set threshold (t), this image will be labelled as blurred and discarded from the sequence. This value is quite subjective depending on the content of the captured scene, however, it has been established via visual analysis of the output that a blur threshold of $t \leq 20\%$ is sufficient for automatically selecting frames that is considered to have minimal blurring and deemed *usable* within the output time-lapse video sequence.

3.3.2 Inactivity frames

A factor that can often arise when capturing a scene over a long period of time is periods of inactivity within the scene, e.g. large time periods where little progress has been made to the construction works within a scene. It is therefore beneficial to determine periods of activity and inactivity so that active frames are only used in the generation of the time-lapse sequence. Due to the camera always having the same viewpoint, comparing the frame that is being processed with the previous frame is useful in detecting inactivity. If the frame has sufficient differences it is deemed as *usable*, if the frame is very similar, the frame will be deemed as *unusable* and will be discarded. The Structural Similarity Index Measure (SSIM) is a perceptual metric that quantifies image quality degradation caused by processing such as data compression or by losses in data transmission. The SSIM index is based on the computation of three terms (luminance, contrast, and the structural term) [Wang et al., 2004]. To detect the differences between two frames, the SSIM (2) is calculated and frames with a very high similarity index will be discarded. Frames with a lower SSIM results in discarding some frames where considerable change is happening. The threshold for this feature is established at 94% in order to acquire sufficient frame labelling results, whilst avoiding the lighting changes and allow small periods of inactivity (false positives) throughout the sequence.

$$\mathrm{SSIM}(\mathrm{x},\mathrm{y}) = \frac{\left(2\mu_\mathrm{x}\mu_\mathrm{y} + \mathrm{C}_1\right) + \left(2\sigma_{\mathrm{xy}} + \mathrm{C}_2\right)}{\left(\mu_\mathrm{x}^2 + \mu_\mathrm{y}^2 + \mathrm{C}_1\right)\left(\sigma_\mathrm{x}^2 + \sigma_\mathrm{y}^2 + \mathrm{C}_2\right)} \quad (2)$$

3.3.3 Day-night detection

In a time-lapse video, the contrast between a frame captured in daylight and another captured at night will have a substantial illumination difference. If the change between day and night is gradual, the time-lapse video will have no issue representing this day-to-night change, however, if there is a dramatic change in frame luminance, e.g. comparison between a day frame directly to a night frame, the resulting transition between frames will be abrupt and result in poor quality time-lapse sequences.

To automate the process of identifying and removing images captured during the night, each frame will be analysed using the pixel-wise average and converted from the RGB colour space to the HSV colour space. This will permit the computation of the average illumination per frame using the brightness matrix (V). Once the luminance is calculated per frame, it is possible to differentiate between day and night as the higher luminance values will represent frames captured during the day and the lower values representing frames captured at night. All night frames will be discarded, resulting in a series of frames that will not flicker due to a large range of luminance.

3.4 Stage III: Frame Enhancement

3.4.1 Detect and remove Timestamp

All the cameras used on this project are utilised for construction management and one important feature of the camera for this task is to print the CCTV on-screen display (OSD) timestamp of the moment that is being recorded. Although this feature is integral for security and monitoring purposes, this feature becomes an issue when using the

 ISBN 978-0-9934207-4-0

footage for the generation of time-lapse videos for marketing purposes. To overcome this, an approach for timestamp removal is proposed to remove this component from each selected frame. To do this, it is first necessary to detect the location where the timestamp is located in each image, i.e. the Region of Interest (ROI). This can be achieved by using a template matching process to locate each of the 18 possible timestamp characters i.e. [*0-9*], "*:*", "-" and [*Mon-Sun*]. This is completed by using a normalised cross correlation template matching approach defined by:

$$\mathrm{R(x,y)} = \frac{\sum x', y'\left(T'(x', y') \cdot I'(x + x', y + y')\right)}{\sqrt{\sum x', y' T'(x', y') 2 \cdot \sum x', y' I'(x + x', y + y') 2}} \tag{3}$$

where *I*=image/frame, *T*=template and *R*=result [Briechle et al., 2001]. The function is convolved with each frame, comparing the overlapped patches $w \times h$ against the desired template shown in Figure 3. The summation is computed over the image patch *x'*=0...*w-1*, *y'*=0...*h-1*. On occasion, a false positive detection of a character can occur in the incorrect location of the image. To alleviate this, an assumption is made that the characters will always be adjacent to each other so the ROI will only be applied when the identified characters are detected in a cluster format. An overall bounding box is applied to the character cluster to represent the acquired ROI (Figure 4).

Figure 3: Mask used during normalised cross correlation template matching

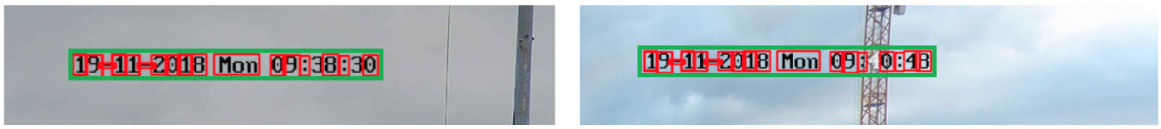


Figure 4: Results obtained from ROI detection

The OSD timestamp font colour (RGB format) will be set by the camera at capture, and will either be black (0,0,0) or white (255,255,255) depending on analysis of the background colour of each scene. Therefore, once the ROI of the timestamp is detected, a binary mask is generated for timestamp removal to accommodate removal of black or white timestamp characters. To accommodate blurred pixel boundaries of the timestamp characters, a range of +/- 30 pixels are used to generate the binary mask. An example binary mask is shown in Figure 5.

Upon generation of a timestamp mask, the inpainting approach from Alexandru Telea [2004] was implemented to reconstruct the pixel-based region where the timestamp is removed. This algorithm starts at the boundary of the ROI and gradually fills, moving towards the ROI centre. Each pixel is replaced by a normalised weighted sum of a 3*x*3 neighbourhood around each pixel. Selection of the weights vary with a greater weighting given to those pixels lying nearer to the normal of the boundary and those lying on the boundary contours. The algorithm is applied to the ROI using the Fast Marching Method.

Figure 5: Generated binary mask used during timestamp removal

3.4.2 Temporal smoothing

To ensure smooth transitions between frames that have been selected for inclusion in a time-lapse video, the Gaussian Mixture Model (GMM) [Reynolds, 2015] was applied. The GMM is a parametric probability density function represented as a weighted sum of Gaussian component densities, where at least three Gaussians are used to model the scene background. Each pixel is assigned a Gaussian mixture to identify the background of a frame. For the purposes of time-lapse frame transition, the model will be applied to every *k* frames, where *k* = (5, 10, 15...)

 ISBN 978-0-9934207-4-0

and use these estimated frames as input frames for the time-lapse video. A background image in the GMM is modelled as:

$$p(x|\lambda) = \sum_{i=1}^{M} w_i g(x|\mu_i, \sigma_i) \quad (4)$$

4 Evaluation

4.1 Algorithmic Performance

To evaluate the performance of the proposed system, the generation of time-lapse videos is compared with the process of generating such time-lapse sequences manually. There are four parameters that influence the performance of time-lapse video generation: Input image resolution; Input video duration; Output time-lapse video duration; Frames/second to be processed. While varying the aforementioned parameters, the acquired processing time and video sequence quality was assessed. Results are presented in Table 1, where the timings for the proposed time-lapse app incorporate the various frame analysis and frame enhancement stages presented in this paper. The manual generation approach is based on the average time a video designer takes to generate a comparable quality time-lapse video. The output time-lapse sequence has a frame rate of 24 fps.

It is evident from the presented results that the proposed automated approach for time-lapse video generation is exponentially more efficient when compared to the manual generation approach. Furthermore, it can be noted that the reduction of image resolution will have a positive impact on the automated generation approach but will have no impact on the manual generation approach, therefore, further highlighting the increased performance that can be obtained via the proposed automated approach.

	Frame Resolution	Manual generation (Human)	Automated generation (Proposed)
Input duration: 1 month Output duration: 1 minute No. of processed frames: 1440	480x270	1 hour 20 minutes	2 minutes 56 seconds
	960x540	1 hour 25 minutes	5 minutes 31 seconds
	1920x1080	1 hour 30 minutes	7 minutes 24 seconds
Input duration: 6 month Output duration: 2 minute No. of processed frames: 2880	480x270	3 hours 40 minutes	4 minutes 36 seconds
	960x540	3 hours 50 minutes	8 minutes 44 seconds
	1920x1080	4 hours 00 minutes	16 minutes 46 seconds
Input duration: 12 month Output duration: 3 minute No. of processed frames: 4320	480x270	7 hours 30 minutes	17 minutes 28 seconds
	960x540	7 hours 45 minutes	32 minutes 51 seconds
	1920x1080	8 hours 00 minutes	41 minutes 42 seconds

Table 1: Comparison of duration between manual and automated method to create time-lapse video sequences

To verify the accuracy of this approach, a number of EverCam Ltd. Customers, who previously received manually generated time-lapse sequences, were provided with auto-generated time-lapse sequences and asked to visually compare the output sequences. The consensus was that the auto-generated video sequences were as visually pleasing as the manually generated sequences, with no obvious differences highlighted.

4.2 FAST Super-Resolution

As shown in Table 1, the processing of low-resolution images considerably decreases the execution time of the proposed algorithm. Therefore, reducing the frame resolution for processing, then increasing the resolution again post processing using the FAST super-resolution methodology, will offer increased processing time, hence substantially improve execution time for time-lapse video generation. For example, if a 1 minute time-lapse video sequence with resolution 1920x1080 can be automatically generated in 12 minutes and 24 seconds, by applying the FAST Super-Resolution [Zhang et al., 2017] algorithm, and applying frame analysis and enhancement on a reduced frame resolution of 480x270, a substantial increase in time for time-lapse video generation can be obtained.

 ISBN 978-0-9934207-4-0

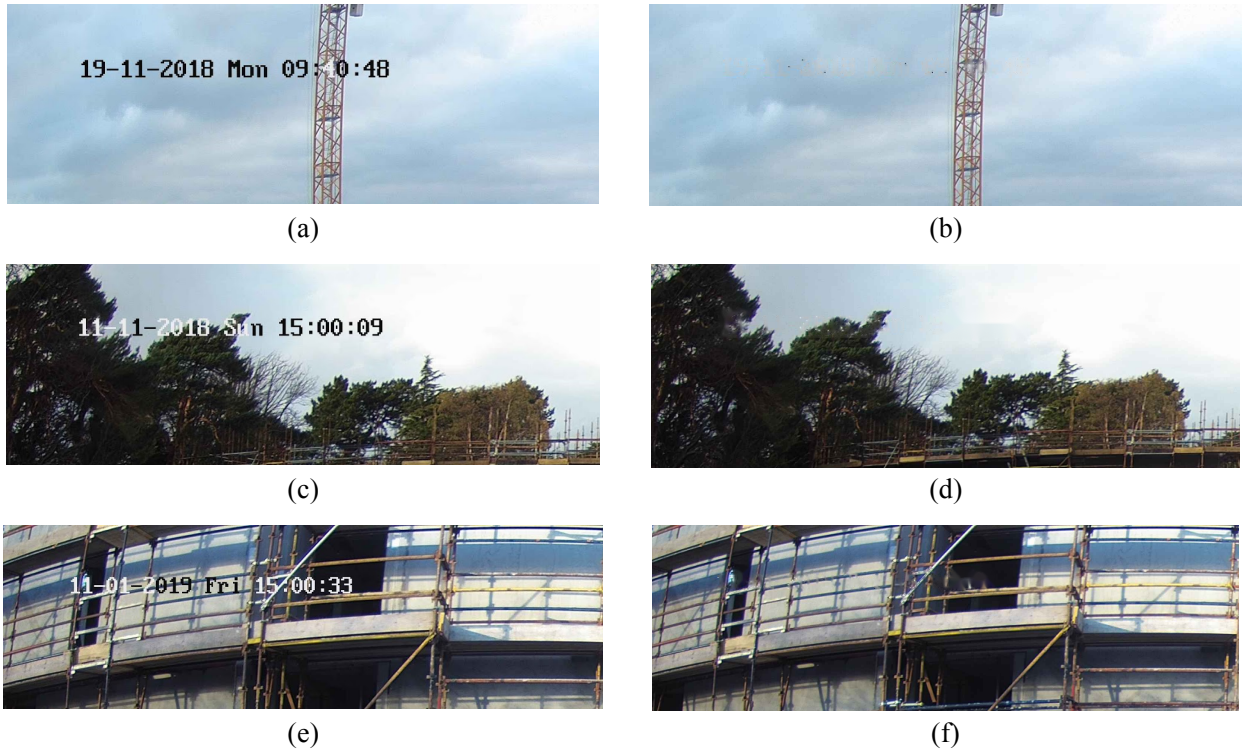
4.3 Frame analysis

Although difficult to comprehend in this paper, the three methods proposed for the analysis of frames, namely frame inactivity, blur detection and day/night detection, has a considerable impact in the generation of a high-quality time-lapse video sequence. When analysing the importance of each method, it is appreciated that the analysis of activity temporally over the captured video stream considerably improves the final time-lapse sequence. For example, a period of inactivity such as holidays on the construction site will generate a period in the time-lapse where no substantial change has occurred. The results obtained by applying frame inactivity analysis have meant that these periods of inactivity are not used in the output time-lapse video sequence, hence producing a higher quality output sequence.

The blur detection and day/night detection features also pay a key part in acquiring a high-quality time-lapse video output. Both approaches offer an automated approach to selecting frames that will enhance the output sequence and disregard any frames that will cause anomalies in the output.

4.4 Detect and remove timestamp

The results obtained when removing the timestamp with the proposed method are considered highly successful. The methodology of first determining the ROI, then applying the timestamp removal to that region offers a robust approach for removing the timestamp from each frame. By visual inspection of the results shown in Figure 6, it can be seen that the automated approach of timestamp removal is success, even in scenes with a lot of variance in the background around the timestamp location.



(a) (b) (c) (d) (e) (f)

Figure 6. Timestamp removed from various construction scenes: (a), (c), (e) represent three originally captured frames with timestamp, (b), (d), (f) represent respective outputs from the timestamp removal process

 ISBN 978-0-9934207-4-0

5 Conclusion

In this paper, an efficient pipeline that will fully automate the generation of high-quality image sequences for high end time-lapse videos fit for company marketing and advertising requirements was presented. The proposed pipeline uses FAST Super-resolution processing to provide an environment that will permit efficient frame processing. Each frame is analysed and classified according to day/night detection, detection of blurred images and activity detection. A novel approach to timestamp removal has been presented and temporal smoothing was used to ensure smooth transition between frames. The obtained results show that the proposed system outperforms manual video editor generation of time-lapse videos by 8 times in terms of processing time, while the quality of the output time-lapse is comparable as a human-made time-lapse video sequence.

6 Acknowledgements

This work is supported in part by Ulster University and IntertradeIreland through the FUSION program, as well as the corporate sponsors Evercam LTD.

7 References

[*Briechle et al.*, 2001] Briechle, K., & Hanebeck, U. D. (2001). *Template matching using fast normalized cross correlation. In Optical Pattern Recognition XII (Vol. 4387, pp. 95-103). Int. Society for Optics and Photonics.*

[Dong et. al., 2015] Dong, C., Loy, C. C., He, K., & Tang, X. (2015). *Image super-resolution using deep convolutional networks*. *IEEE transactions on pattern analysis and machine intelligence*, *38*(2), 295-307.

[Joshi et. al., 2015] Joshi, N., Kienzle, W., Toelle, M., Uyttendaele, M., & Cohen, M. F. (2015). *Real-time hyperlapse creation via optimal frame selection. ACM Transactions on Graphics (TOG), 34(4), 63.*

[Koppal and Narasimham, 2006] Koppal, S. J., and Narasimham, S. G. (2006) *Clustering appearance for scene analysis. In Proc. of Computer Vision Pattern Recognition Conference, vol. 2, 1323 – 1330.*

[Kupyn et al., 2004] Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., & Matas, J. (2018). *Deblurgan: Blind motion deblurring using conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 8183-8192)*.

[Matusik et al., 2004] Matusik, W., Loper, M., Pfister, H. (2004). *Progressively-refined reflectance functions from natural illumination. In Rendering Techniques, Eurographics Association, Keller and H. W. Jensen, Eds., 299–308.*

[Moey, 2019] Moey Inc. (2019). new york timelapse — Moey Inc.. [online] *Available at: http://moeyinc.com/new-york-timelapse [Accessed 30 May 2019]*.

[Radford et. al., 2015] Radford, A., Metz, L., & Chintala, S. (2015). *Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.*

[Reynolds, 2015] Reynolds, D. (2015). *Gaussian mixture models. Encyclopedia of biometrics, 827-832.*

[Tao et. al., 2017] Tao, X., Gao, H., Liao, R., Wang, J., & Jia, J. (2017). *Detail-revealing deep video super-resolution. In Proceedings of the IEEE International Conference on Computer Vision (pp. 4472-4480).*

[Telea, 2004] Telea, A. (2004). *An image inpainting technique based on the fast marching method. Journal of graphics tools, 9(1), 23-34.*

[Wang et al., 2004] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). *Image quality assessment: From error measurement to structural similarity. IEEE trans. Image Processing, 13(1).*

[Xiong et. al., 2018] Xiong, W., Luo, W., Ma, L., Liu, W., & Luo, J. (2018). *Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2364-2373)*.

[Zhang et al., 2017] Zhang, Z., & Sze, V. (2017). *FAST: A framework to accelerate super-resolution processing on compressed videos. IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 19-28).*